Hindawi Publishing Corporation Journal of Computational Engineering Volume 2014, Article ID 175820, 6 pages http://dx.doi.org/10.1155/2014/175820



Research Article

An Improved Unscented Particle Filter with Global Sampling Strategy

Yi-zheng Zhao

Hefei Electronic and Engineering Institute, Hefei 230000, China

Correspondence should be addressed to Yi-zheng Zhao; yi6666@163.com

Received 13 July 2014; Accepted 22 November 2014; Published 10 December 2014

Academic Editor: Hongli Dong

Copyright © 2014 Yi-zheng Zhao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Particle filter (PF) has many variations and one of the most popular is the unscented particle filter (UPF). UPF uses the unscented Kalman filter (UKF) to generate particles in the PF framework and has a better performance than the standard PF. However, UPF suffers from its high computation complexity because it has to execute UKF to each particle to obtain proposal distribution. This paper gives an improved UPF aiming at reducing the computation complexity of the algorithm. In comparison to the standard UPF, the new strategy generates proposal distribution from the mean and covariance value of the whole particles instead of from each particle. Thus the improved algorithm utilizes the characteristics of the whole particles and only needs to perform UKF algorithm once to get the proposal distribution at each time step. Experimental results show that, compared to standard UPF, the improved algorithm reduces the time consumption greatly almost without performance degradation.

1. Introduction

Nonlinear and non-Gaussian filtering has a wide range of applications in many fields [1–3]. Among the many methods that have been proposed in the literature for these applications, particle filter (PF) has become one of the most popular. For decades PF has been applied to a variety of problems, such as computer vision, signal processing [4], target tracking [5], and financial pricing [6].

However when designing a PF a major problem is to choose a proper proposal distribution of the particles. Due to the fact that the particles are drawn from this distribution, and the weight values of particles are also related to this distribution, the performance of a PF is strongly influenced by the choice of the proposal distribution.

To design better proposal distributions, several techniques based on linearization have been proposed. In one method, an extended Kalman filter (EKF) is used to generate the proposal distribution, and this is known as extended Kalman PF (EKPF). However, the linearization operation in EKPF introduces modeling errors, which can yield large estimation errors if the system is highly nonlinear [7]. To overcome this problem, a more accurate PF was proposed using

unscented Kalman filter (UKF) to generate the proposal distribution [8]. The UKF can accurately compute the mean and covariance of nonlinear systems up to the second order of the Taylor series expansions. This type of particle filter using UKF to generate proposal distribution is known as the unscented particle filter (UPF). The UKF produces proposal distributions that exhibit a larger support overlap with the true posterior probability than the EKF proposal distributions; thus it is more suitable for the proposal distribution generation, and consequently the UPF performs much better than EKPF [9].

However, in UPF framework, an unscented Kalman filtering operation which is often more complex to compute needs to be applied to each particle to generate and propagate a Gaussian proposal distribution. Generally there are many particles in particle filter; therefore the time complexity of UPF is often tens of times larger than general PF; therefore the application of UPF is limited in many occasions.

In order to reduce the time cost of the UPF, Fasheng and Yuejin [10] proposed an IUPF method that divides the particles into two parts, one part generated from the transition prior and another part from the UKF. Basically this method is a combination of the general PF and the UPF. Because only

part of the particles needs to be generated using UKF, this method has a less time cost than UPF. For the same reason, however, with less particles generated from UKF, the filter accuracy is also decreased compared to the UPF. Thus IUPF method can hardly keep high accuracy and low computation simultaneously.

To reduce the computation complexity of the UPF and maintain the advantage of its high precision, this paper focuses on an improved UPF which generates proposal distribution from the mean and covariance of the whole particles instead of from each particle. Stimulation experiments show that, by this strategy, the time cost of UPF can be reduced greatly and almost without loss of accuracy.

The rest of this paper is organized as follows. In Section 2, we briefly reviewed the general particle filter algorithm and the UPF. Section 3 gives the details of the proposed particle filter. Experiments and analysis are given in Section 4. The last section draws the conclusion.

2. Particle Filter

2.1. General Particle Filter. The nonlinear, non-Gaussian filtering problem considered here consists of recursively computing the posterior probability density function of the state vector in a general discrete-time state-space model, given the observed measurements. Such a general model can be formulated as

$$\mathbf{x}_{t} = \mathbf{f} \left(\mathbf{x}_{t-1}, \mathbf{w}_{t-1} \right),$$

$$\mathbf{y}_{t} = \mathbf{h} \left(\mathbf{u}_{t}, \mathbf{x}_{t}, \mathbf{v}_{t} \right),$$
(1)

where $\mathbf{y}_t \in R^{n_y}$ denotes the output observations, $\mathbf{u}_t \in R^{n_u}$ denotes the input observations, $\mathbf{x}_t \in R^{n_x}$ denotes the state of the system, $\mathbf{w}_t \in R^{n_w}$ denotes the process noise, and $\mathbf{v}_t \in R^{n_v}$ denotes the measurement noise. The mappings $\mathbf{f}: R^{n_x} \times R^{n_w} \mapsto R^{n_x}$ and $\mathbf{h}: R^{n_x} \times R^{n_u} \mapsto R^{n_y}$ represent the deterministic process and measurement models. The prior distribution at t = 0 is represented by $p(\mathbf{x}_0)$.

The main idea underlying the particle filter is to approximate the posterior probability density distribution $p(\mathbf{x}_{0:k} \mid \mathbf{y}_{1:k})$ by using a set of random samples with associated weights, which are called "particles." The weight assigned to each particle is proportional to the probability. Let $\{\mathbf{x}_{0:k}^i, \widetilde{\omega}_k^i\}_{i=1}^N$ denote a random measure that characterizes the posterior probability density distribution $p(\mathbf{x}_{0:k} \mid \mathbf{y}_{1:k})$, and $\{\mathbf{x}_{0:k}^i, i=1,\ldots,N\}$ is a set of particles with their associated weights $\{\widetilde{\omega}_k^i, i=1,\ldots,N\}$. The weights are normalized as $\sum_{i=1}^N \widetilde{\omega}_k^i = 1$, where N is the number of samples used in the approximation. The posterior density at time k can be approximated as

$$p\left(\mathbf{x}_{0:k} \mid \mathbf{y}_{1:k}\right) \approx \sum_{i=1}^{N} \widetilde{\omega}_{k}^{i} \cdot \delta\left(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{i}\right). \tag{2}$$

The normalized importance weights $\tilde{\omega}_k^i$ are chosen by using the principle of importance sampling. If the samples $\mathbf{x}_{0:k}^i$ are drawn from a so-called proposal distribution

 $p(\mathbf{x}_{0:k} | \mathbf{y}_{1:k})$ with the principle of Sequential Importance Sampling, $\widetilde{\omega}_k^i$ can be given by

$$\widetilde{\omega}_k^i = \frac{\omega_k^i}{\sum_{i=1}^N \omega_k^j},\tag{3}$$

where

$$\omega_{k}^{i} \propto \frac{p\left(\mathbf{x}_{0:k}^{i} \mid \mathbf{y}_{1:k}\right)}{q\left(\mathbf{x}_{0:k}^{i} \mid \mathbf{y}_{1:k}\right)} = \omega_{k-1}^{i} \frac{p\left(\mathbf{y}_{k} \mid \mathbf{x}_{k}^{i}\right) p\left(\mathbf{x}_{k}^{i} \mid \mathbf{x}_{k-1}^{i}\right)}{q\left(\mathbf{x}_{k}^{i} \mid \mathbf{x}_{0:k-1}^{i}, \mathbf{y}_{1:k}\right)}.$$
 (4)

Equation (4) provides a mechanism to sequentially update the importance weights, given an appropriate choice of proposal distribution, $q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})$. The choice of proposal distribution is one of the most important issues for particle filter [6]. The standard particle filter uses the transition prior $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ as proposal density [7], which means

$$q\left(\mathbf{x}_{k} \mid \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}\right) = p\left(\mathbf{x}_{k} \mid \mathbf{x}_{k-1}\right). \tag{5}$$

By substitution of (5) into (4), the weights can be recursively computed and updated as

$$\omega_k^i \propto \omega_{k-1}^i p\left(\mathbf{y}_k \mid \mathbf{x}_k^i\right). \tag{6}$$

It is worth to notice that if the samples are generated from a suboptimal proposal such as in (5), the weights of most of them will approach zero after a few iterations. Therefore the filter spends a considerable amount of computational time in updating unlikely samples. Generally this can be avoided by the resampling step, which consists in replacing the unlikely samples with the more likely ones [8].

A better proposal distribution can help solve this problem with no need to resample particles, which is discussed in the next section.

2.2. UPF. In order to overcome weights degeneracy of general particle filter, the unscented Kalman filter is adopted as the proposal density function in general particle filter, which is called UPF. It has been proven that the UKF is a better proposal than the transition prior in that it incorporates the most recent observations which usually contain much valuable information for estimating states [9].

In comparison to other linearization strategies, UPF tends to generate more accurate estimates of the true covariance of the state. Unlike the EKFP, this method does not linearize the nonlinear equations of the system, and consequently it can accurately capture the nonlinearity of the system. Details of the UPF algorithm can be found in [10]. Figure 1 gives the main steps of UPF in a flow chart.

3. Improved Unscented Particle Filter

However, in UPF framework, unscented Kalman filtering operation is used to generate and propagate a Gaussian proposal distribution to each particle, which leads to a huge computational complexity. In this paper, an improved strategy based on global sampling UPF (GS-UPF) is proposed to

reduce computation complexity. Before generating proposal distribution at time k, the new strategy calculates the mean $\overline{\mathbf{x}}_{k-1}$ and covariance $\widehat{\mathbf{P}}_{k-1}$ of all particles at time k-1, which is then used in unscented Kalman filter to update the approximate mean of state $\overline{\mathbf{x}}_k$ and covariance $\widehat{\mathbf{P}}_k$ with the latest measure y_k . After that, the Gaussian distribution $N(\overline{\mathbf{x}}_k, \widehat{\mathbf{P}}_k)$ is constructed as the proposal distribution and N particles are drawn from it. To sum up, instead of applying UKF to each particle, this new strategy only needs one UKF operation at each time step and consequently reduces computation. Moreover, since there is no weight degeneration in the new algorithm, thus the new strategy does not have to take any resample operation. The details of GS-UPF are as follows.

Step 1 (initialization: t = 0). Draw particles $\{\mathbf{x}_0^i\}_{i=1}^N$ from the prior $p(\mathbf{x}_0)$.

Step 2. Calculate the mean and covariance of particles at time k-1:

$$\mathbf{x}_{k-1} = \sum_{i=1}^{N} \widetilde{\omega}_{k-1}^{i} \mathbf{x}_{k-1}^{i}$$

$$\widehat{\mathbf{P}}_{k-1} = \sum_{i=1}^{N} \widetilde{\omega}_{k-1}^{i} \left(\mathbf{x}_{k-1}^{i} - \overline{\mathbf{x}}_{k-1} \right) \left(\mathbf{x}_{k-1}^{i} - \overline{\mathbf{x}}_{k-1} \right)^{T}.$$
(7)

Step 3. Use UKF to predict and update the mean and covariance of particles at time k-1.

(a) Extend the state vectors:

$$\mathbf{x}_{k-1}^{a} = \begin{bmatrix} \overline{\mathbf{x}}_{k-1} & w_{k-1} & v_{k-1} \end{bmatrix}^{T}$$

$$\mathbf{P}_{k-1}^{a} = \begin{bmatrix} \widehat{\mathbf{P}}_{k-1} & 0 & 0 \\ \mathbf{0} & \mathbf{Q}_{k-1} & \mathbf{0} \\ 0 & 0 & \mathbf{R}_{k-1} \end{bmatrix}.$$
(8)

(b) Calculate Sigma points:

$$\boldsymbol{\chi}_{k-1}^{a} = \left[\overline{\mathbf{x}}_{k-1}^{a} \ \overline{\mathbf{x}}_{k-1}^{a} \pm \left(\sqrt{\left(n_{a} + \lambda \right) \mathbf{P}_{k-1}^{a}} \right) \right]. \tag{9}$$

(c) Calculate the weights:

$$W_0^{(m)} = \frac{\lambda}{n_x + \lambda}, \qquad W_0^{(c)} = \frac{\lambda}{n_x + \lambda} + (1 - \alpha^2 + \beta)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{\lambda}{2(n_x + \lambda)}, \quad i = 1, 2, \dots, 2n_x.$$
(10)

(d) Time update:

$$\chi_{k|k-1}^{x} = \mathbf{f}\left(\chi_{k-1}^{x}, \chi_{k-1}^{w}\right), \quad \overline{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n_a} W_i^{(m)} \chi_{i,k|k-1}^{x}$$

$$\mathbf{P}_{\mathbf{x}_{k|k-1}\mathbf{x}_{k|k-1}} = \sum_{i=0}^{2n_a} W_i^{(c)} \left(\boldsymbol{\chi}_{i,k|k-1}^x - \overline{\mathbf{x}}_{k|k-1} \right) \left(\boldsymbol{\chi}_{i,k|k-1}^x - \overline{\mathbf{x}}_{k|k-1} \right)^T$$

$$\boldsymbol{\gamma}_{k|k-1} = \mathbf{h}\left(\boldsymbol{\chi}_{k|k-1}^{x}, \boldsymbol{\chi}_{k|k-1}^{v}\right), \qquad \widehat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2n_a} W_i^{(m)} \boldsymbol{\gamma}_{i,k|k-1},$$

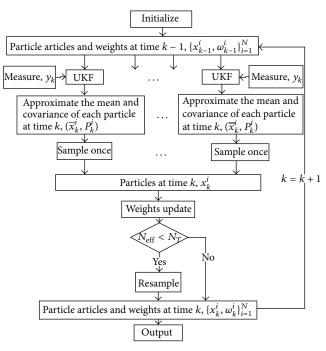


FIGURE 1: Flow diagram of UPF.

where $\chi_{i,k-1}^a$ is the *i*th column vector of χ_{k-1}^a ; $\gamma_{i,k|k-1}$ is the *i*th column vector of $\gamma_{k|k-1}$.

(e) Measurement update:

$$\mathbf{P}_{\mathbf{y}_{k|k-1}\mathbf{y}_{k|k-1}} = \sum_{i=0}^{2n_{a}} W_{i}^{(c)} \left(\mathbf{y}_{i,k|k-1} - \widehat{\mathbf{y}}_{k|k-1} \right) \left(\mathbf{y}_{i,k|k-1} - \widehat{\mathbf{y}}_{k|k-1} \right)^{T} \\
\mathbf{P}_{\mathbf{x}_{k|k-1}\mathbf{y}_{k|k-1}} = \sum_{i=0}^{2n_{a}} W_{i}^{(c)} \left(\mathbf{\chi}_{i,k|k-1}^{x} - \widehat{\mathbf{x}}_{k|k-1} \right) \left(\mathbf{y}_{i,k|k-1} - \widehat{\mathbf{y}}_{k|k-1} \right)^{T} \\
\mathbf{K}_{k} = \mathbf{P}_{\mathbf{x}_{k|k-1}\mathbf{y}_{k|k-1}} \mathbf{P}_{\mathbf{y}_{k|k-1}\mathbf{y}_{k|k-1}}^{-1}, \\
\overline{\mathbf{x}}_{k} = \overline{\mathbf{x}}_{k|k-1} + \mathbf{K}_{k} \left(\mathbf{y}_{k} - \widehat{\mathbf{y}}_{k|k-1} \right) \\
\widehat{\mathbf{P}}_{\mathbf{x}_{k|k}\mathbf{x}_{k|k}} = \mathbf{P}_{\mathbf{x}_{k|k-1}\mathbf{x}_{k|k-1}} - \mathbf{K}_{k} \mathbf{P}_{\mathbf{y}_{k|k-1}\mathbf{y}_{k|k-1}} \mathbf{K}_{k}^{T}. \tag{12}$$

Step 4. Importance sampling is as follows:

- (a) sample \mathbf{x}_k^i from $N(\mathbf{x}_k; \overline{\mathbf{x}}_k, \widehat{\mathbf{P}}_k)$;
- (b) weight update: $\omega_k^i = \omega_{k-1}^i(p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)p(\mathbf{y}_k \mid \mathbf{x}_k^i)/N(\mathbf{x}_k^i; \overline{\mathbf{x}}_k, \widehat{\mathbf{P}}_k))$, and normalize the weights.

Step 5. Output particles with weights, and get the state estimation.

Step 6. If the next observation comes, set k = k + 1, and go to Step 2; otherwise, exit the algorithm.

Figure 2 shows these steps in a flow chart.

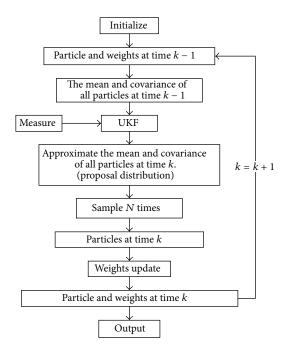


FIGURE 2: Flow diagram of GS-UPF.

4. Simulation Experiments

4.1. Experiment Model. In order to evaluate the efficiency of the improved UPF algorithm presented above, several experiments are carried out. A typical nonlinear process model and measurement model are as follows:

$$x_k = 1 + \sin(0.4\pi k) + 0.5x_{k-1} + w_{k-1}$$

$$y_k = \begin{cases} 0.2x_k^2 + v_k, & k \le 30\\ 0.5x_k + v_k - 2, & k > 30, \end{cases}$$
(13)

where w_k is a Gamma distribution (3, 0.5) random variable modeling the process noise, and the measurement noise v_k is drawn from a Gaussian distribution N(0, 0.01).

The parameters of the models are $x_0 = 1$, $P_0 = 3/4$, $\alpha = 1$, $\beta = 0$, and $\kappa = 2$, and the number of effective samples $N_{\rm eff}$ is N/3. The number of Monte Carlo simulations is MC = 100, and the time step T = 60.

The root of mean square error (RMSE) of each independent run is defined as

RMSE =
$$\left(\frac{1}{T}\sum_{i=1}^{T} (\hat{x}_i - x_i)^2\right)^{1/2}$$
. (14)

4.2. Results Analysis. Figure 3 shows the measurement, true state, and state estimation of UPF and GS-UPF, from which it can be found that both of the algorithms can estimate the true state effectively and there is only some slight difference. Figure 4 gives the estimation error of UPF and GS-UPF; it can also be found that the estimation error of the two algorithms is almost identical to each other.

Figure 5 shows the RMSE comparison of EKPF, UPF, and GS-UPF, from which we can find that RMSE of GS-UPF is

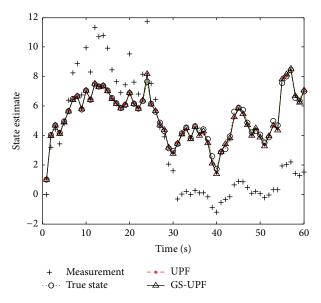


FIGURE 3: State estimation of UPF and GS-UPF (N = 300).

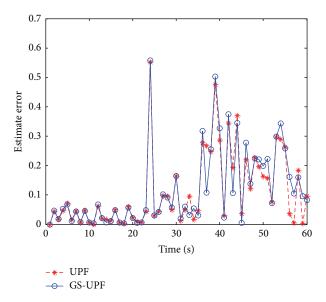


FIGURE 4: Estimation error of UPF and GS-UPF (N = 300).

slightly larger than that of UPF and both of them is much smaller than RMSE of EKPF. It means the GS-UPF has almost the same filter accuracy with the UPF, and both are much more accurate than EKPF. From Figure 5 it can be found that the RMSE of three algorithms keeps decreasing as the particle number increases, and along with the increase of the particle number, the trend of decreasing of RMSE slows down.

Figure 6 gives the average time consumption of EKPF, UPF, and GS-UPF, which is defined as Time = $(1/\text{MC})\sum_{j=1}^{\text{MC}}\sum_{k=1}^{T}\text{Time}_{k}^{j}$, where Time_{k}^{j} is the time cost at time k of state estimation in jth Monte Carlo simulation. It is shown that the UPF costs most time while GS-UPF has a much less time consumption, which is even less than EKPF. From Figure 6 we can also find that, compared to UFP, GS-UPF reduces the

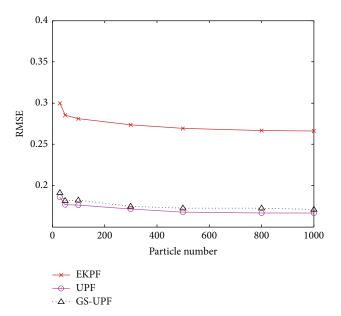


FIGURE 5: RMSE of EKPF, UPF, and GS-UPF at different particle numbers.

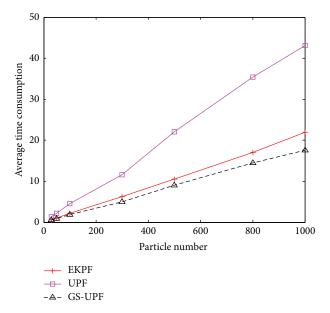


FIGURE 6: Average time consumption of EKPF, UPF, and GS-UPF.

time consumption more than 50% to 60% at various particle numbers.

The proposed algorithm is also compared with the IUPF method presented in [11]. In IUPF, the proportion of the particles generated from UKF is c=20%, 50%, and 80%, and the other particles are generated from the transition prior, respectively. The algorithms use 300 particles. The comparison on the time cost and accuracy of the UPF, GS-UPF, and IUPF is presented in Table 1. The GS-UPF is more accurate than IUPF in all conditions. The time cost is also less than IUPF when c=50% and 80%. When c=20%, IUPF is

TABLE 1: Comparison between GS-UPF and the IUPF.

	Time cost (seconds)	RMSE
UPF	11.36	0.15
GS-UPF	4.98	0.17
IUPF, $c = 20\%$	3.49	0.35
IUPF, $c = 50\%$	5.42	0.26
IUPF, $c = 80\%$	7.81	0.21

faster while its precision is seriously lower than GS-UPF. Thus it fails to keep high precision of UPF when *c* is relatively small.

In summary, from above experimental results, we can find that the GS-UPF can greatly reduce the computation complexity and maintain high accuracy of UPF at the same time.

5. Conclusion

By applying an improving strategy to UPF, this paper aims at reducing the time consumption of UPF and adopts its main merits at the same time. The new proposed GS-UPF uses the characteristics of the particle set and uses UKF algorithm only once at each time step to get a Gaussian distribution as a proposal distribution in the improved algorithm. This strategy reduces lots of computation because it does not have to apply UKF to each particle as the UPF does. The experimental results indicate that, compared to UPF, the GS-UPF keeps high filter accuracy and reduces much time consumption, making it more practicable in the nonlinear areas applications.

Conflict of Interests

The author declares being the only author of this paper and having no conflict of interests.

References

- [1] H. Dong, Z. Wang, and H. Gao, "Distributed H_{∞} filtering for a class of markovian jump nonlinear time-delay systems over lossy sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, pp. 4665–4672, 2013.
- [2] H. Dong, Z. Wang, and H. Gao, "Distributed filtering for a class of time-varying systems over sensor networks with quantization errors and successive packet dropouts," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 3164–3173, 2012.
- [3] Z. Wang, H. Dong, B. Shen, and H. Gao, "Finite-horizon H_{∞} filtering with missing measurements and quantization effects," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1707–1718, 2013.
- [4] Y. Yoshida, K. Hayashi, H. Sakai, and W. Bocquet, "Marginalized particle filter for blind signal detection with analog imperfections," *IEICE Transactions on Communications*, vol. 93, no. 2, pp. 336–344, 2010.
- [5] F. Wang, Q. Guo, and Y. Lin, "A Kalman particle filter for bearing-only target tracking," *Journal of Computational Information Systems*, vol. 7, no. 15, pp. 5628–5635, 2011.

- [6] D. Creal, "A survey of sequential Monte Carlo methods for economics and finance," *Econometric Reviews*, vol. 31, no. 3, pp. 245–296, 2012.
- [7] O. Cappe, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.
- [8] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," Tech. Rep. CUED/F-INFENG/TR 380, Department of Engineering, Cambridge University, 2000.
- [9] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
- [10] W. Fasheng and L. Yuejin, "Improving particle filter with a new sampling strategy," in *Proceedings of the 4th International* Conference on Computer Science and Education, pp. 408–412, July 2009.
- [11] F. Wang, Y. Lin, T. Zhang, and J. Liu, "Particle filter with hybrid proposal distribution for nonlinear state estimation," *Journal of Computers*, vol. 6, no. 11, pp. 2491–2501, 2011.

















Submit your manuscripts at http://www.hindawi.com























